ITERATIVE ANALYSIS OF DOCUMENT COLLECTIONS ENABLES EFFICIENT HUMAN-INITIATED INTERACTION

Joseph CHAZALON, Bertrand COÜASNON

Rennes, Brittany, France

www.irisa.fr/intuidoc

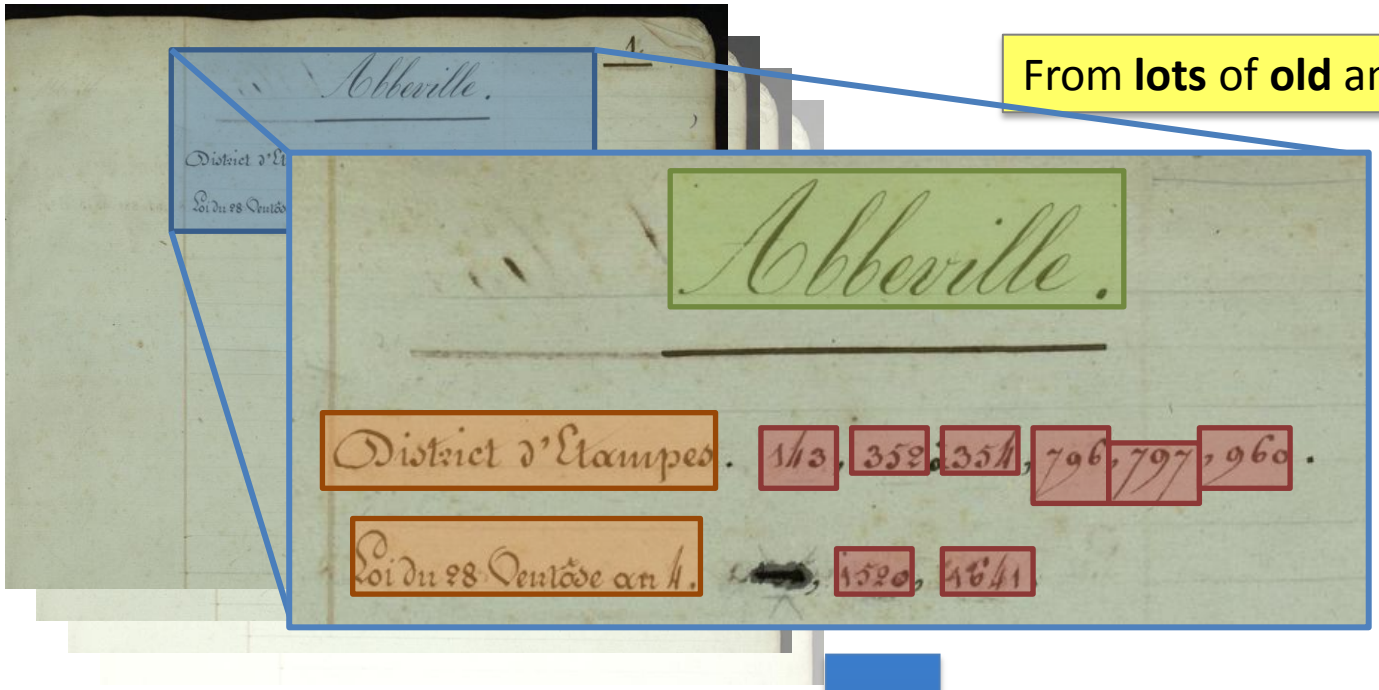
Project funding

Employers

Research unit

Regional Research & Education Network

We extract, recognize and index contents.



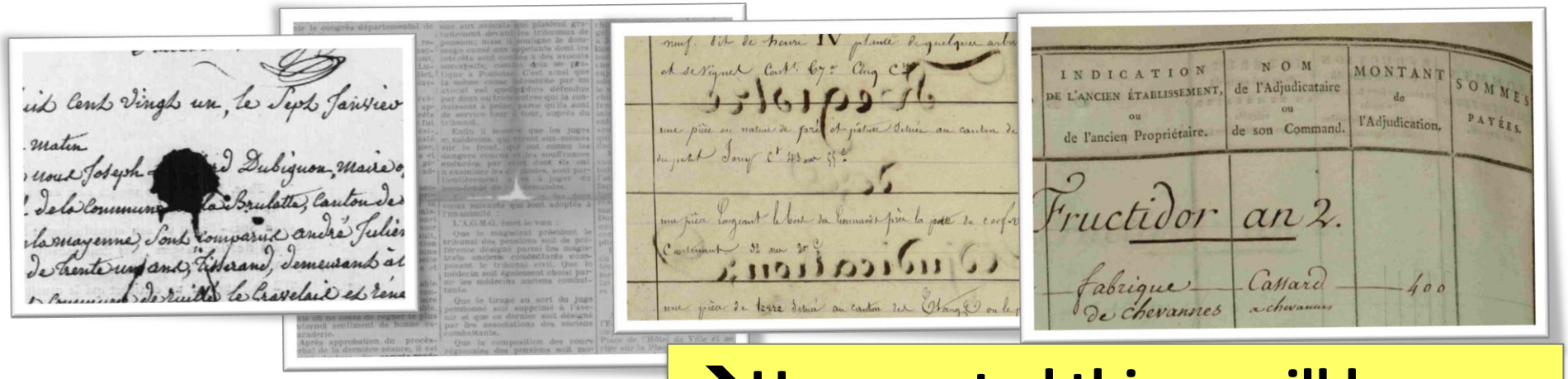
From **lots** of **old** and **damaged** images…

…to **database**.

Image file	Town / Place	Kind of sale	Sale reference
00071.jpg	Abbeville	dist_etampes	143
00071.jpg	Abbeville	loi_ventose	1520

Challenges for degraded document analysis

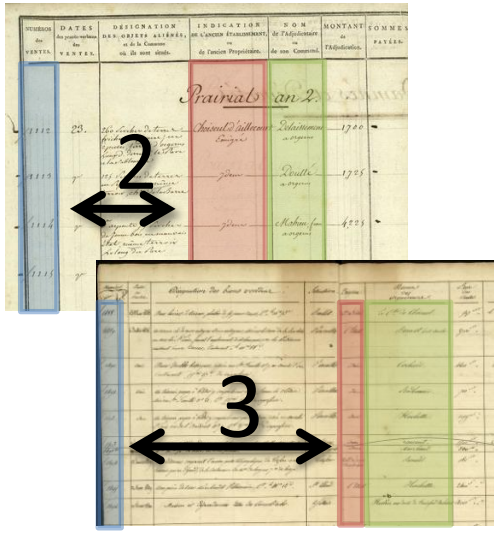
Alterations

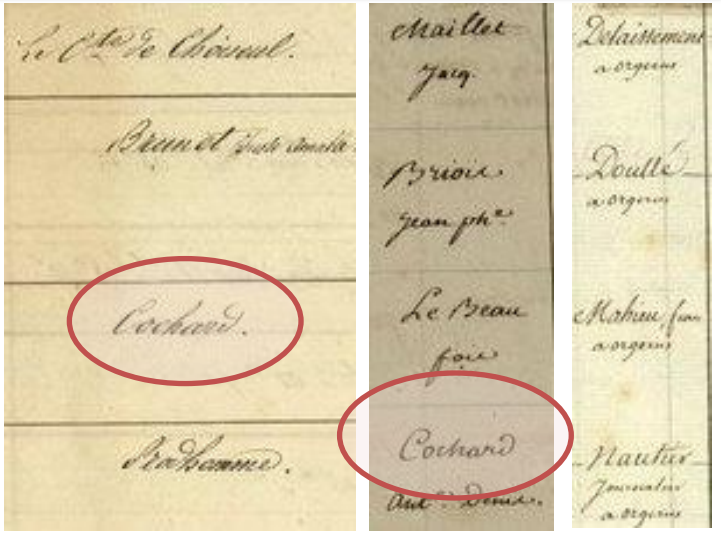


➔**Unexpected things will happen**

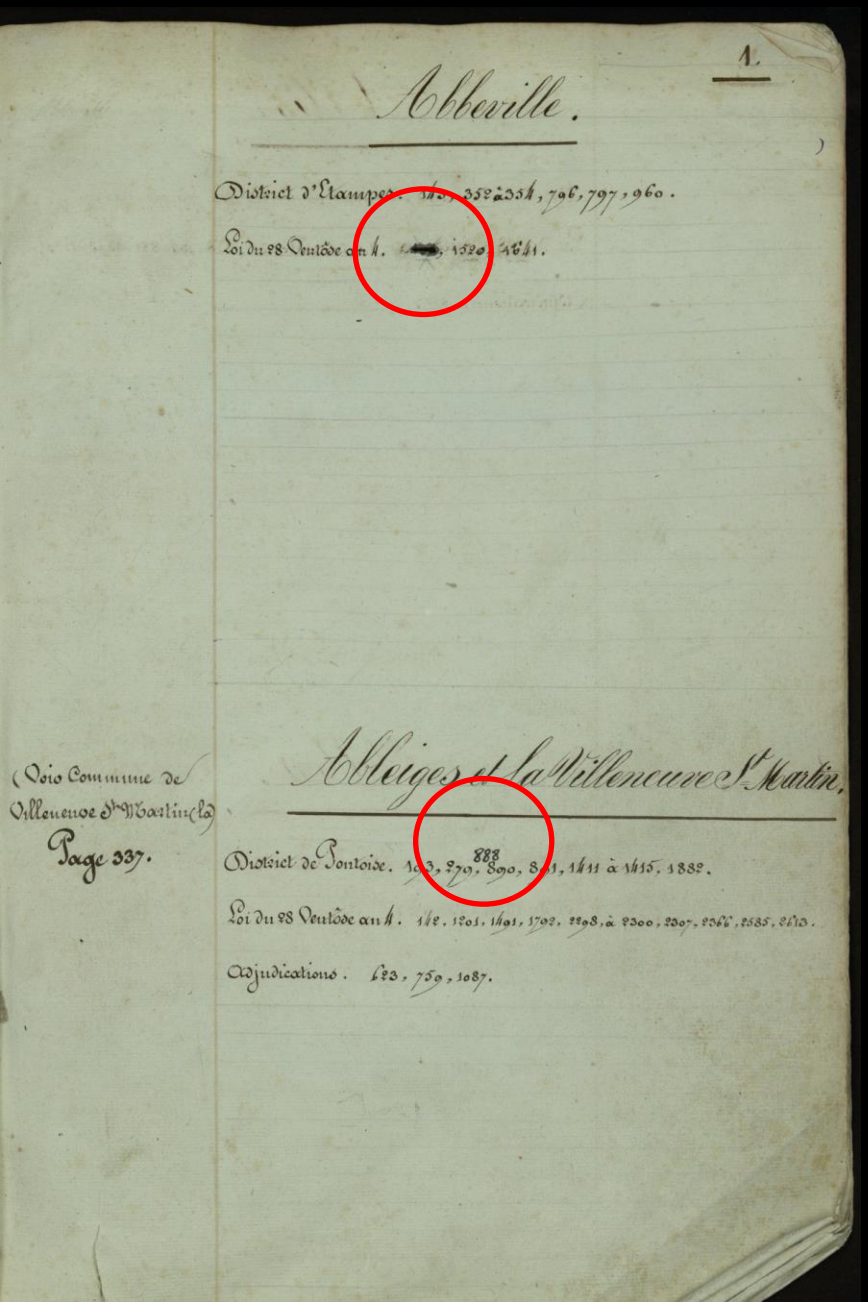
➔**Human help is required**

Variability





Example: 18th century documents



1.

Abbeville.

District d'Etampes. [illegible], 352, 354, 796, 797, 960.

Loi du 28 Ventôse an 4. [illegible], 1520, 1641.

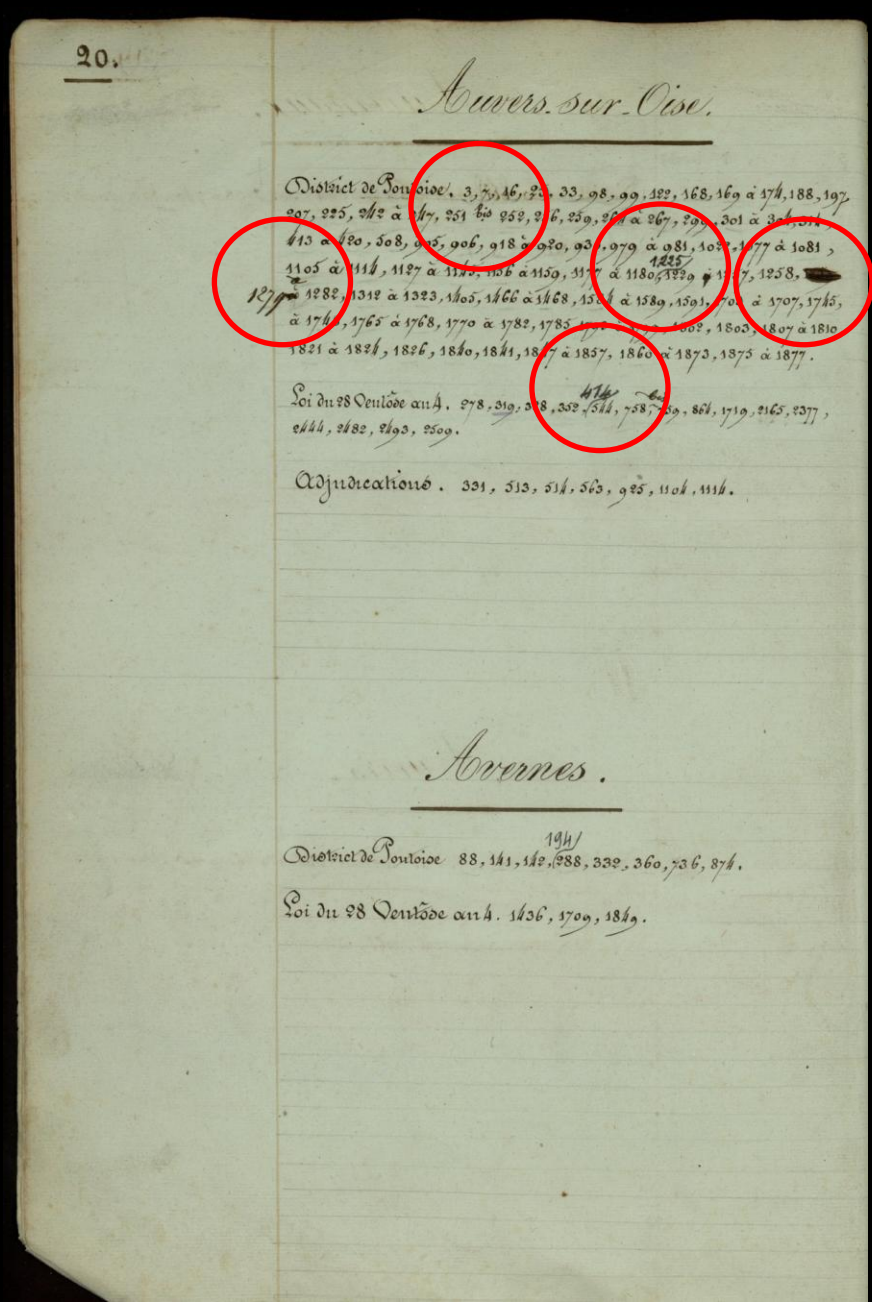
Abloiges et la Villeneuve St Martin.

(Voir Commune de Villeneuve St Martin (la) Page 337.

District de Pontoise. 1[illegible]3, 270, 888, 890, 8[illegible]1, 1411 à 1415, 1882.

Loi du 28 Ventôse an 4. 142, 1201, 1491, 1702, 2298, à 2300, 2307, 2546, 2585, 2610.

Adjudications. 423, 759, 1087.



20.

Auvers sur Oise.

District de Pontoise. 3, 7, 16, 32, 33, 98, 99, 122, 168, 169 à 174, 188, 197, 207, 225, 242 à 247, 251 bis 252, 256, 259, 266 à 267, 299, 301 à 304, 311, 413 à 420, 508, 905, 906, 918 à 920, 930, 979 à 981, 1037, 1077 à 1081, 1105 à 1114, 1127 à 1135, 1156 à 1159, 1177 à 1180, 1225 1229 [illegible] 1257, 1258, [illegible] 1279 1282, 1312 à 1323, 1465, 1466 à 1468, 1584 à 1589, 1591, 1702 à 1707, 1745, à 1764, 1765 à 1768, 1770 à 1782, 1785 [illegible], 1803, 1807 à 1810, 1821 à 1824, 1826, 1840, 1841, 1857 à 1857, 1860 à 1873, 1875 à 1877.

Loi du 28 Ventôse an 4. 278, 319, 348, 382, 514, 758, [illegible], 861, 1719, 2165, 2377, 2444, 2482, 2493, 2509.

Adjudications. 331, 513, 514, 563, 925, 1104, 1114.

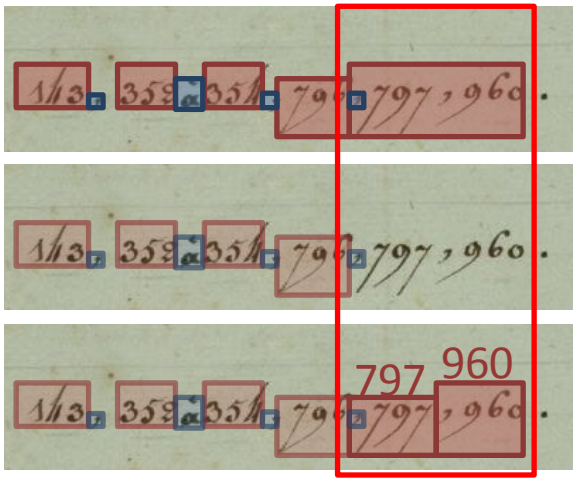
Avernes.

District de Pontoise 88, 141, 142, 194 1888, 339, 360, 736, 874.

Loi du 28 Ventôse an 4. 1436, 1709, 1849.

Example: Handling under-segmentations

Usually: correct errors **during post-processing**

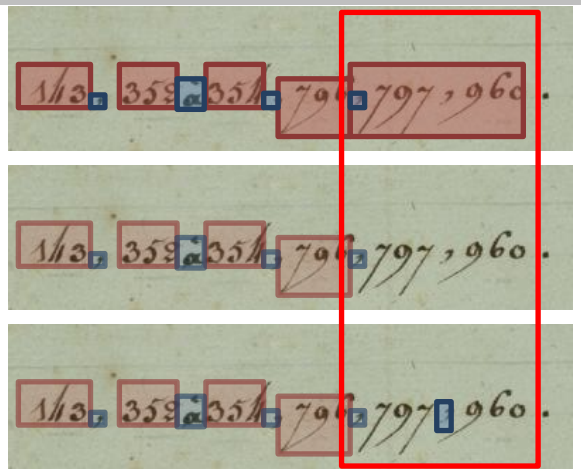


1. Locate under-segmentation in analysis results
2. Clear zone
3. Add a zone for each number, (and key its value)

How is the final structure regenerated?
(town x sale_kind x sale_reference)

Costly operation

Our approach: correct errors **during the analysis**

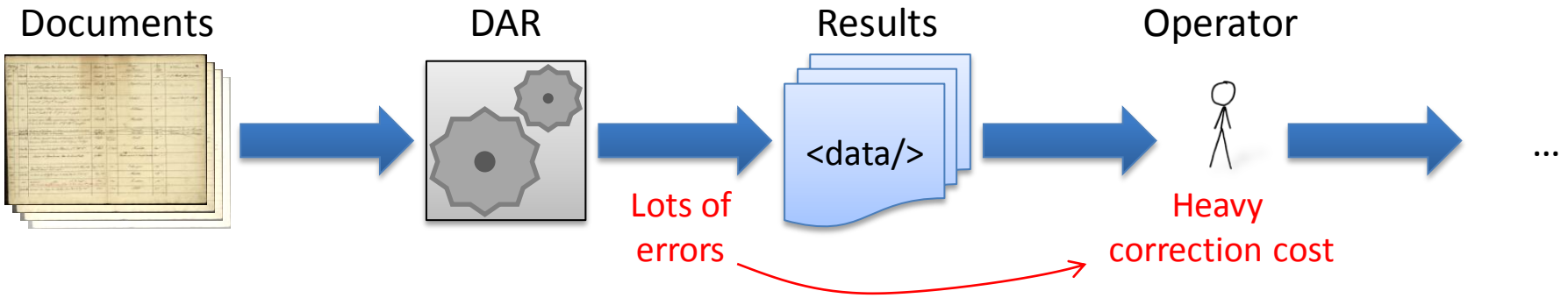


1. Locate under-segmentation in analysis results
2. Clear zone
3. **Add a missing separator** (pen stroke?)

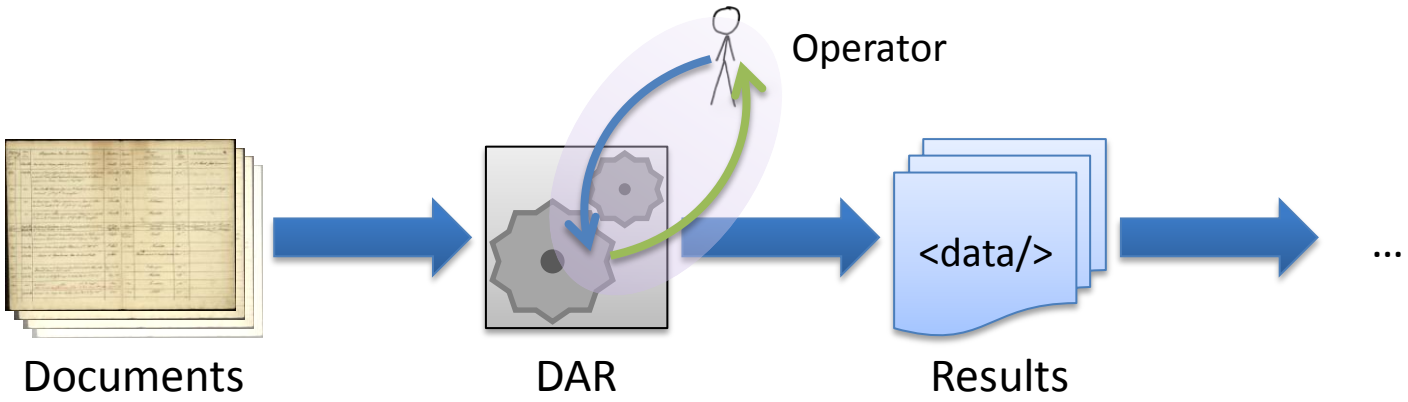
Number detection and recognition, final structure generation : ***automatic***

Benefit of human interaction during analysis

Post-processing: complex and costly



During the analysis: Lighter correction cost?



- ✓ Early error correction
- ✓ Limited error propagation

Constraints for human interaction

How to interact? Be efficient.

- External information must be used to **improve responses**
- Interaction must be **asynchronous**
  - prevent the human and the system from waiting for each other

When interaction should be triggered?

- As soon as a problem happens
  - Easy if **automatic error detection** is possible (ask a question)
  - But **otherwise** ?

Today's topic

Interaction models

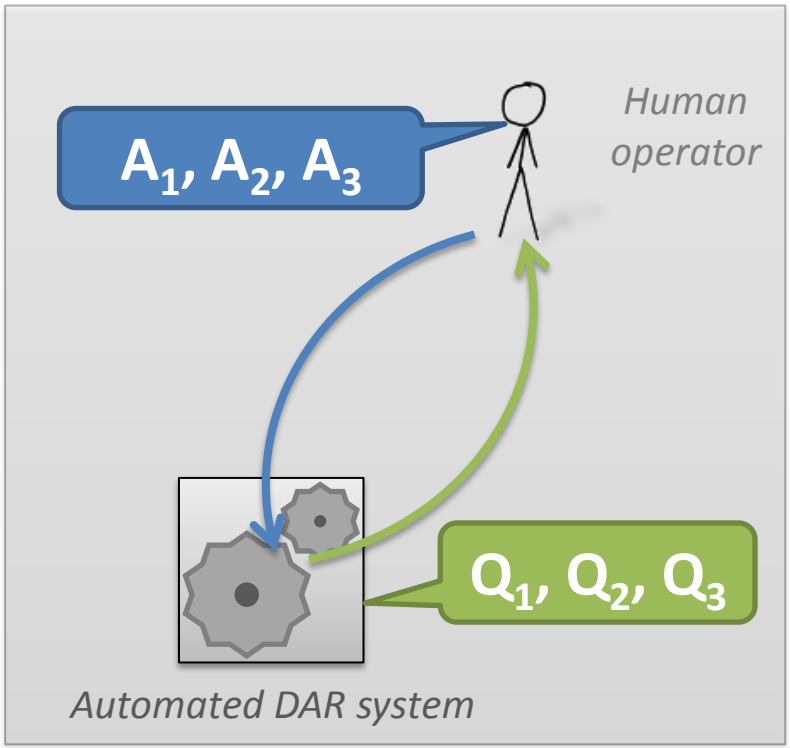
Directed interaction *[ICDAR 2011]*

[system] detects errors automatically

[system] asks questions

[human] answers **each** questions

[system] uses the answers to progress



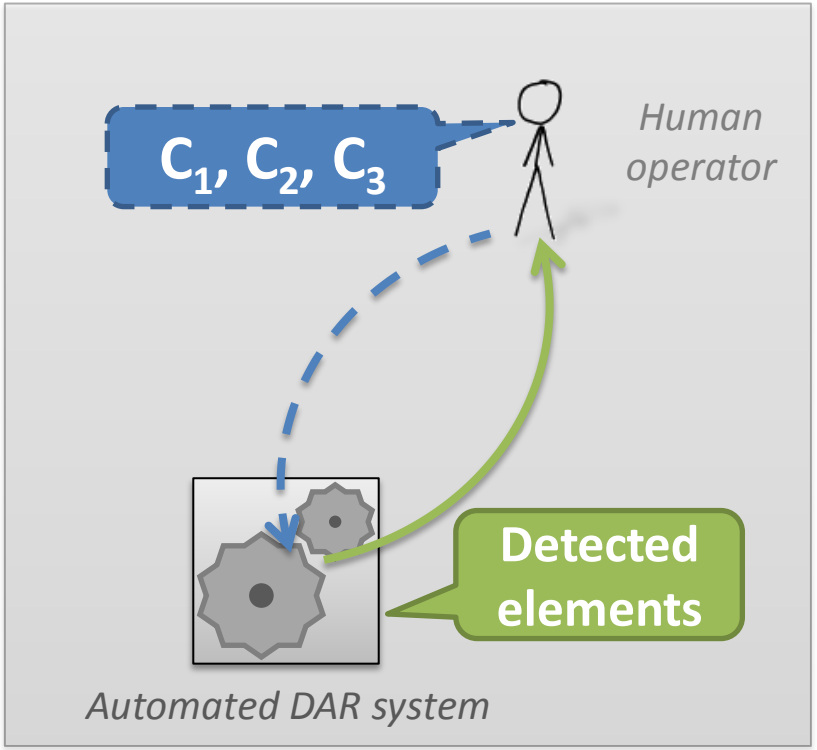
Spontaneous interaction

Today's topic

[system] shows detected elements

[human] makes correction**s**

[system] uses external information (**if possible**)



Presentation outline

1. **System architecture** for an iterative analysis

   Enabling a spontaneous interaction

2. **Implementation** of an iterative page analyzer

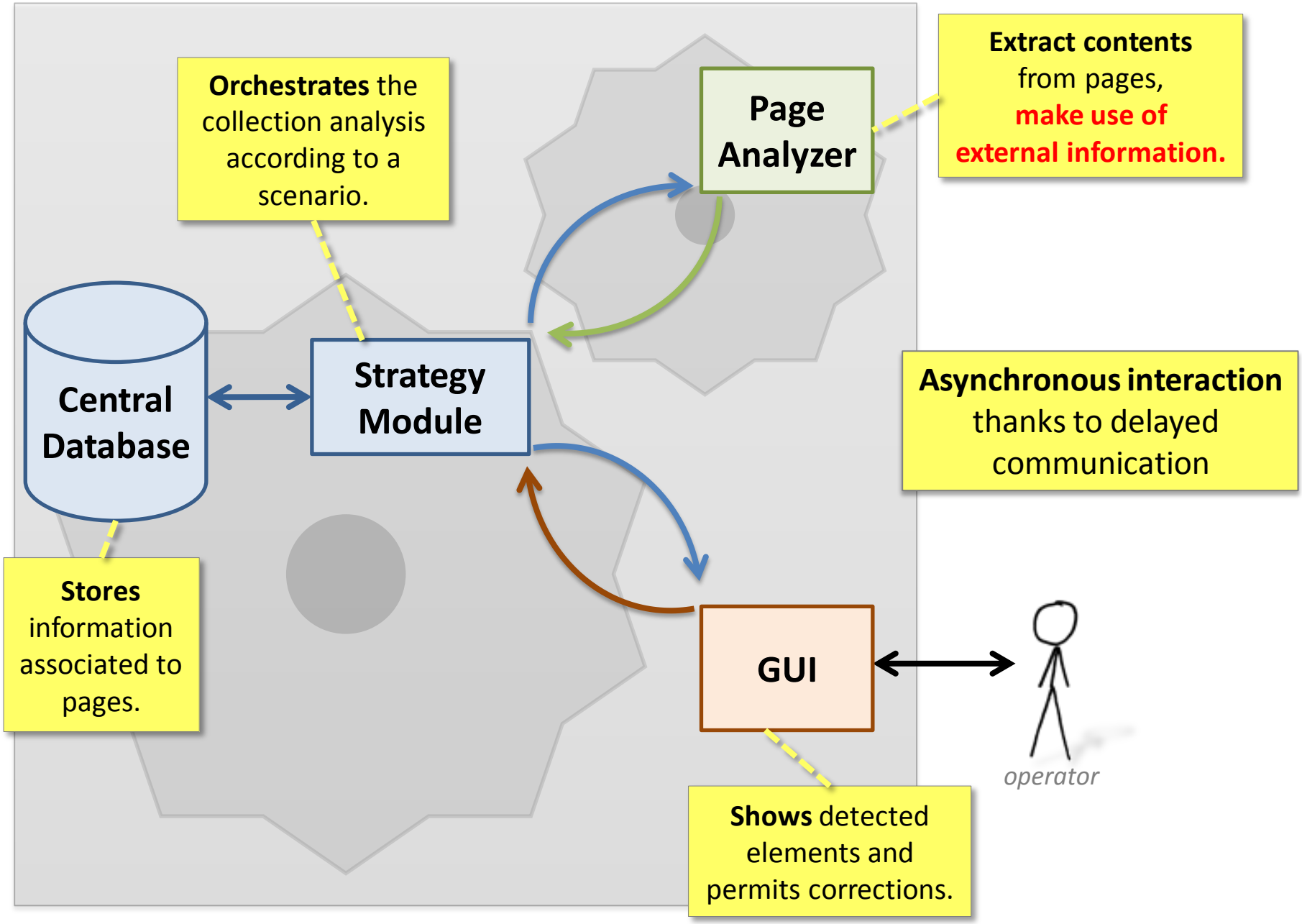
   Using external information during page analysis

3. **Experiments** and **results**

Enabling a **spontaneous** interaction

SYSTEM ARCHITECTURE FOR AN ITERATIVE ANALYSIS

Required components



Iterative analysis of our documents

*Step 1: Automated analysis of **all pages***

1

Process page P_1

Page Analyzer

2

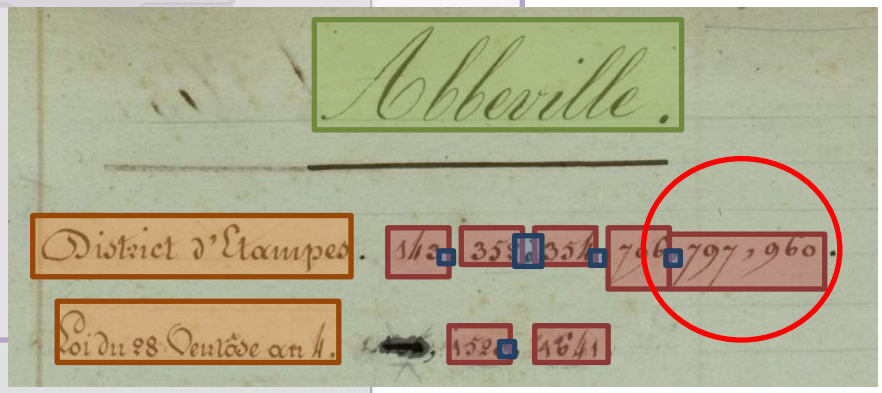
Here are the elements I detected…

Central Database

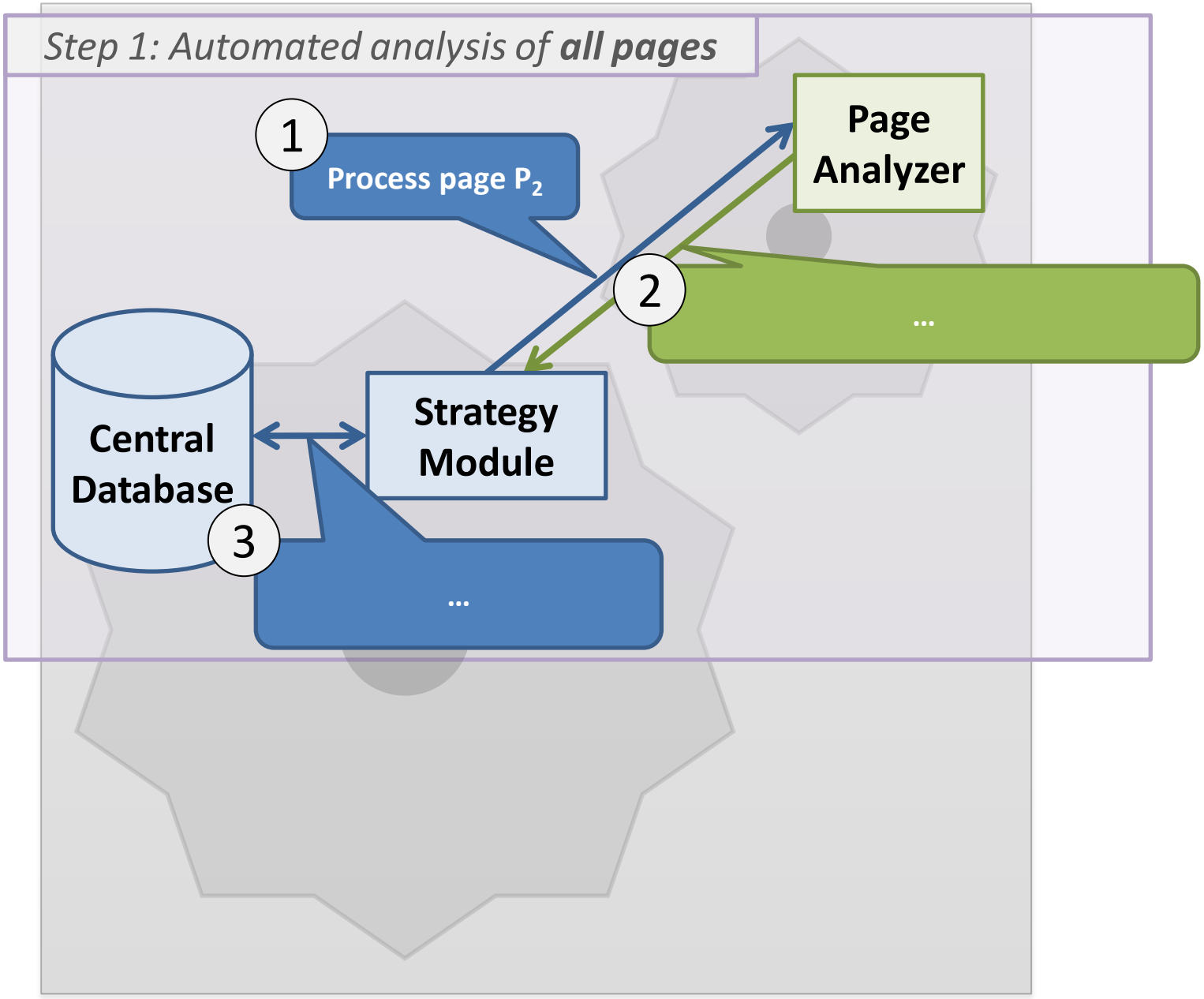
Strategy Module

3

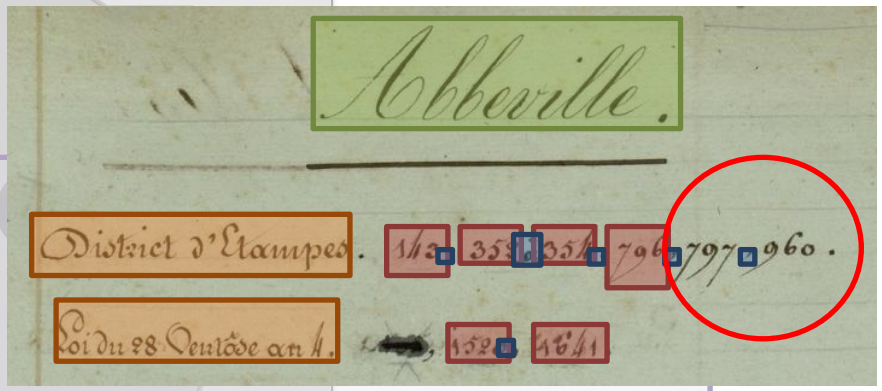
Store new data for P_1



Iterative analysis of our documents



Iterative analysis of our documents



Step 2: Interactive session

1 Get data for P_1

Central Database

Strategy Module

2 Show elements detected

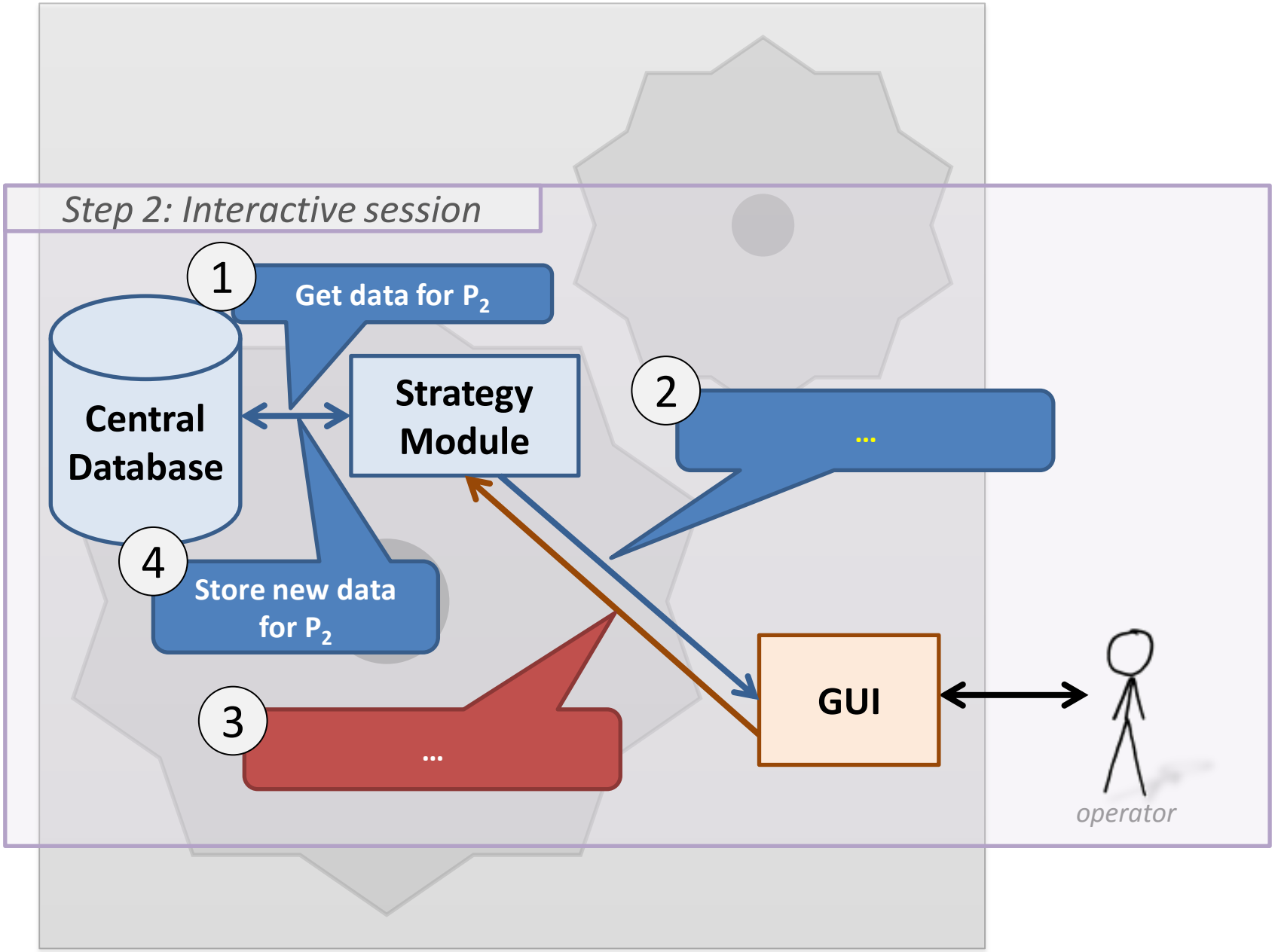
4 Store new data for P_1

3 Corrections

GUI

operator

Iterative analysis of our documents



Iterative analysis of our documents

*Step 3: Reprocessing of **all pages***

2 Process page P_1
PS: Here are some user corrections

1 Get data for P_1

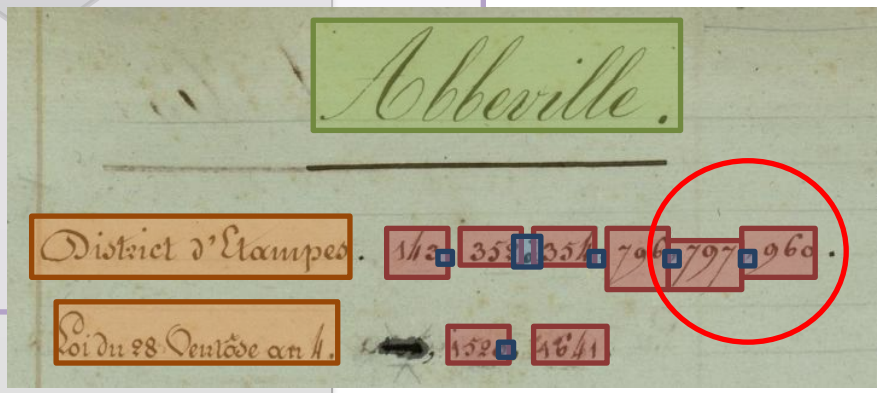
Page Analyzer

3 Here are the elements I detected…

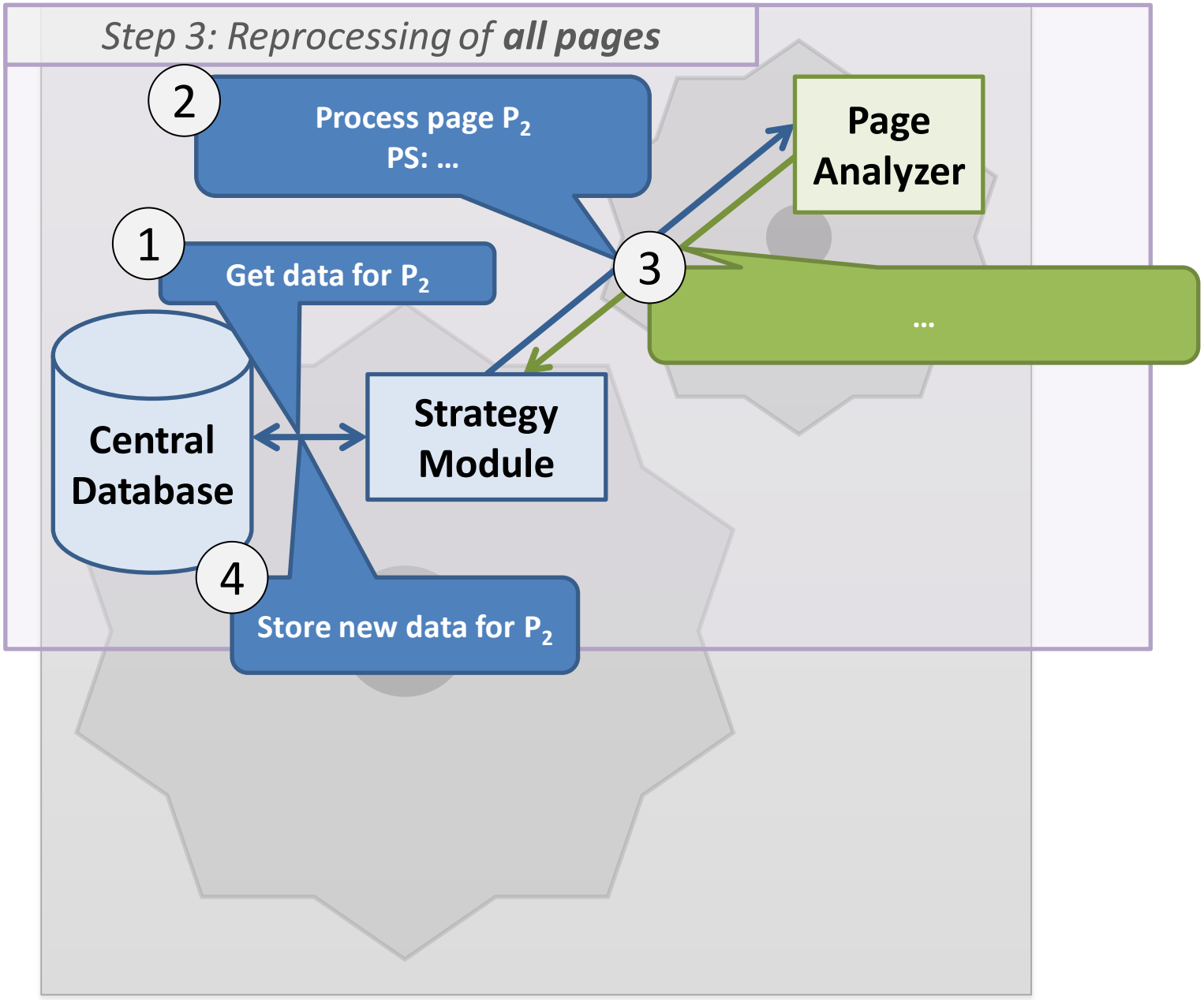
Central Database

Strategy Module

4 Store new data for P_1



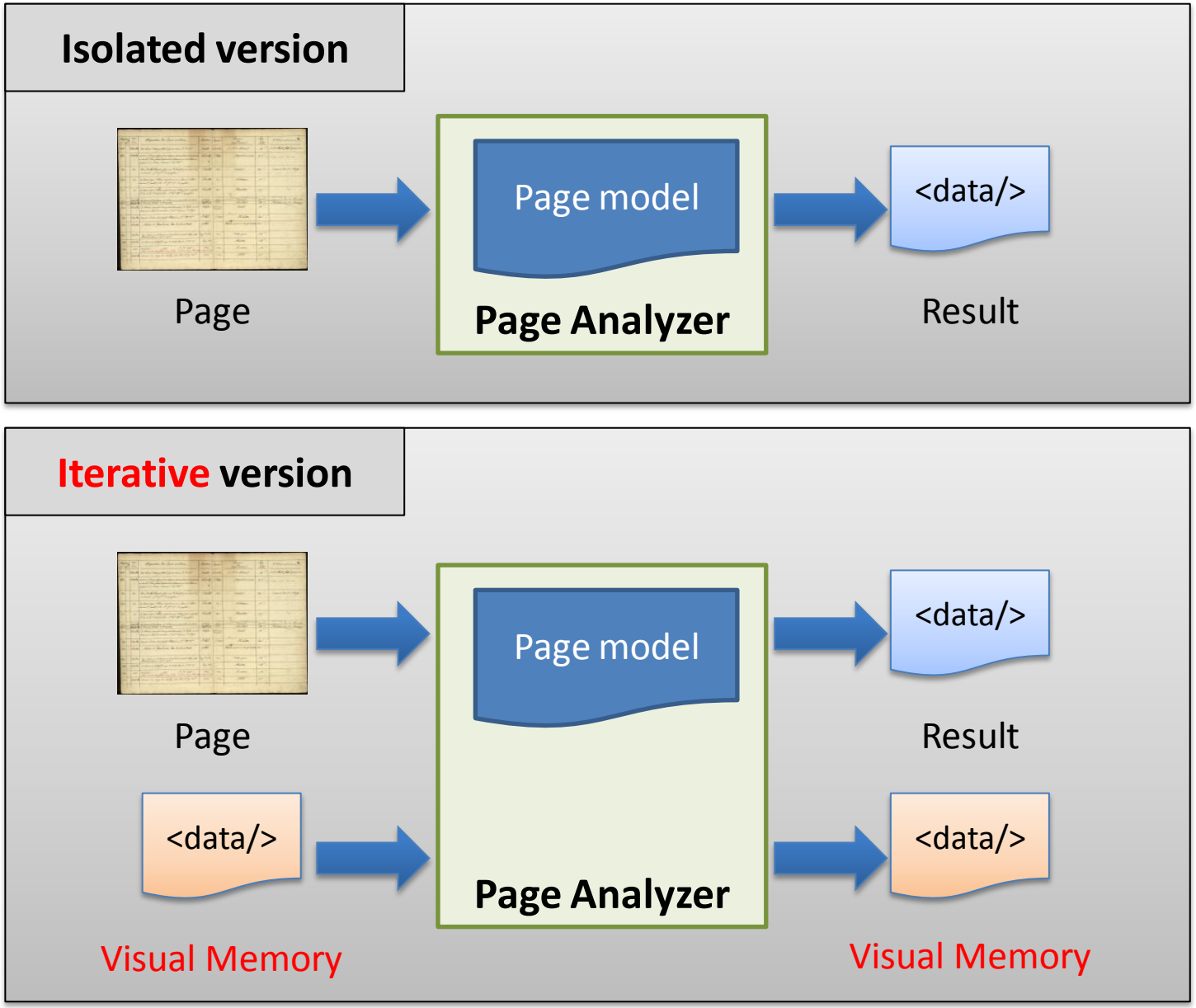
Iterative analysis of our documents



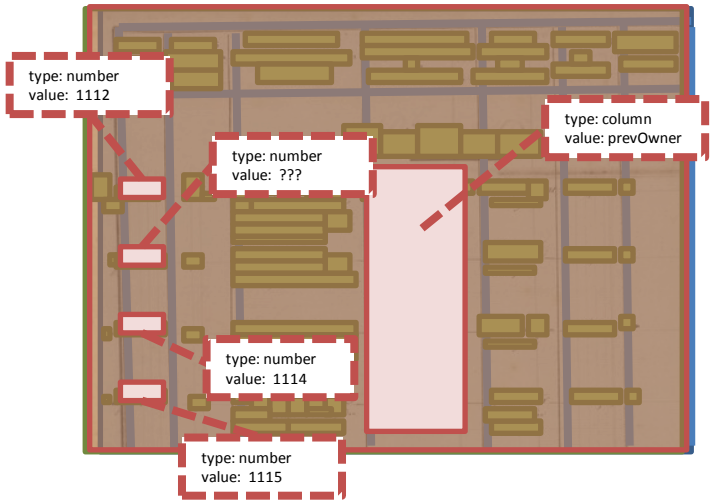
Using external information
during page analysis

IMPLEMENTATION OF AN ITERATIVE PAGE ANALYZER

Focusing on the Page Analyzer



Visual Memory & Page Analyzer

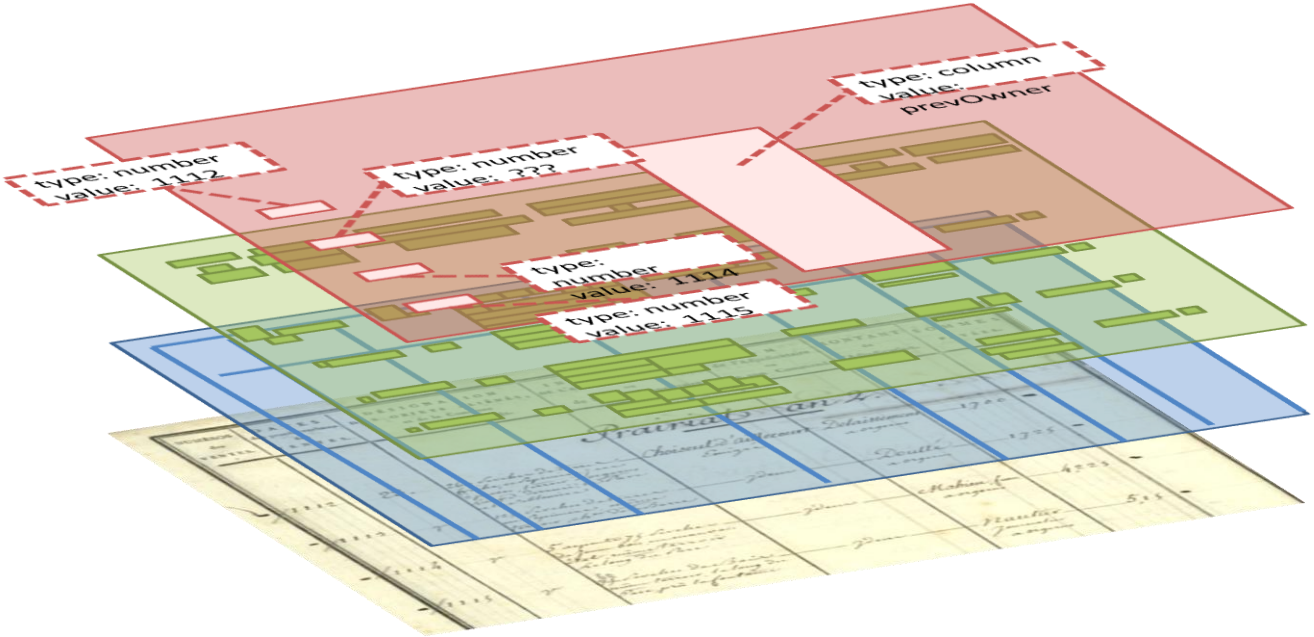


We use a layered structure for page analysis

External information uses one more layer

This *Visual Memory* has 3 properties:

1. Same referential as the image
   each element has a shape + a position
2. Information is available at any moment
3. Same access operations as image data
   easy creation, modification, deletion



External data = Visual Memory

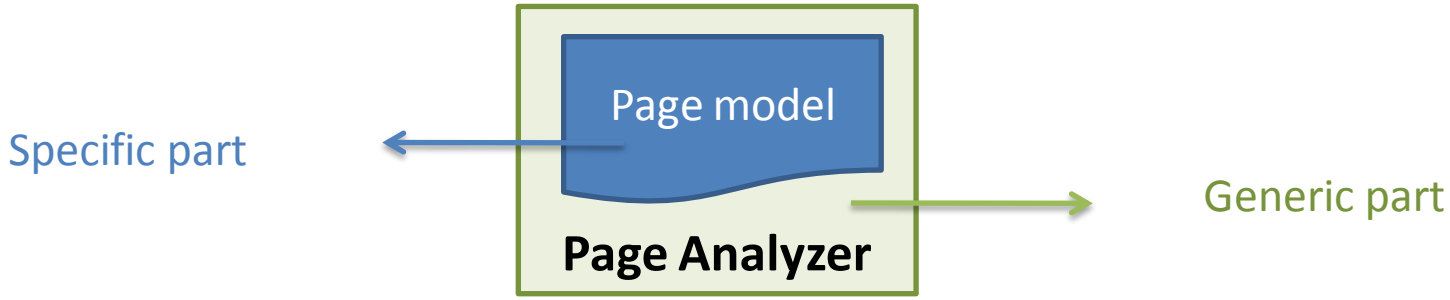
Components

Lines

Image

Merging external and image information

Page analyzer base design



Grammatical description

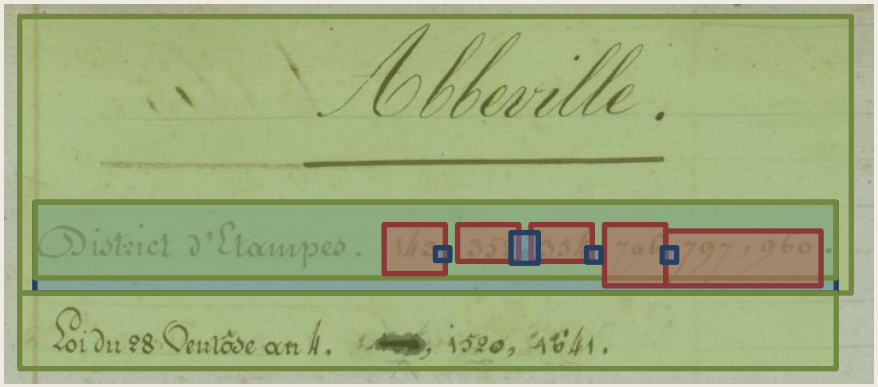
- entities to locate and recognize
- relative and absolute positioning
- other properties with precise semantics

Analyzer's behavior associated to operators

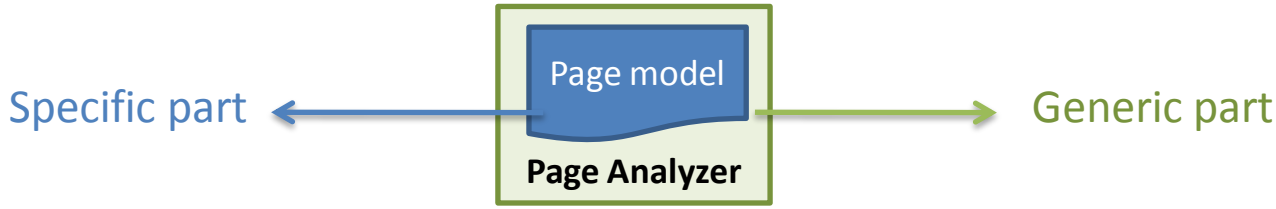
- analysis is guided by the description
- operators implementation rule the analysis

```
start() ::=
   AT(topPage) &&
   detectTextLinePosition(-Line) &&
   IN(lineArea +Line)
   DO(extractSalesNumbers()) &&
   AT(under +Line) &&
   % read other lines...

extractSalesNumbers() ::=
   detectAllSeparators(-SepLst) &&
   extractNumberBetweenSep(+SepLst).
```



New operator for spontaneous interaction



New operator

Tags at part of the description

Semantics: associated subpart of the analysis can be performed externally

```
SPONTANEOUS(marker, Rule)
```

Associated behavior

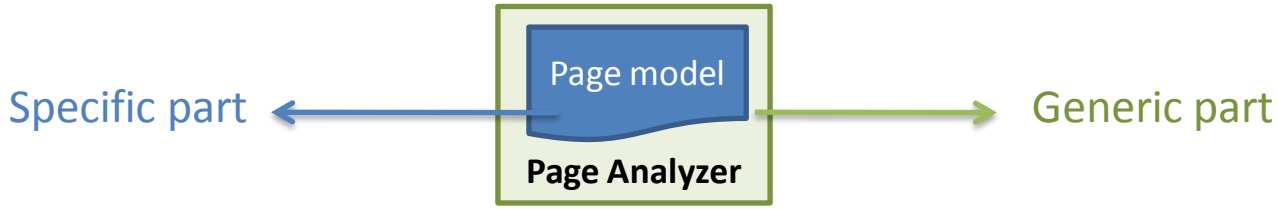
If some element (with the right marker) exists in the visual memory of the image
Then
use this element as a result and return
Else
perform the usual analysis

Without interaction

```
detectAllSeparators(-[Sep|OtherSep]) ::=
   separator(-Sep) &&
   detectAllSeparators(-OtherSep).

detectAllSeparators(-[]).
```

New operator for spontaneous interaction



New operator

Tags at part of the description

Semantics: associated subpart of the analysis can be performed externally

```
SPONTANEOUS(marker, Rule)
```

Associated behavior

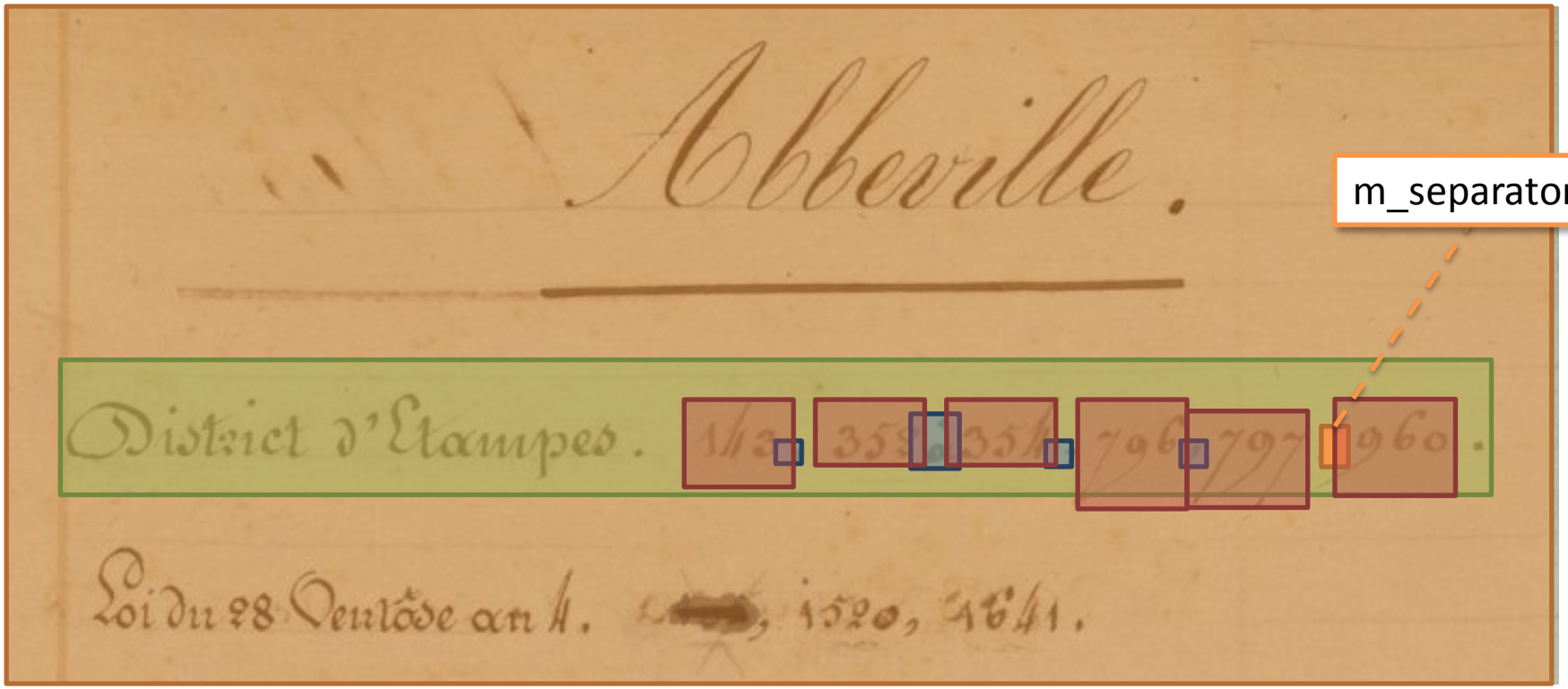
If some element (with the right marker) exists in the visual memory of the image
Then
use this element as a result and return
Else
perform the usual analysis

With interaction

```
detectAllSeparators(-[Sep|OtherSep]) ::=
    SPONTANEOUS(+m_separator, separator(-Sep)) &&
    detectAllSeparators(-OtherSep).

detectAllSeparators(-[]).
```

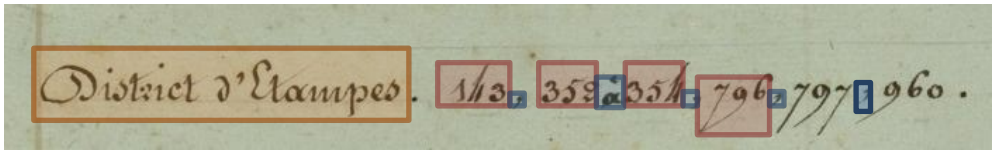
New operator for spontaneous interaction



- New **information source**: visual memory
- New separators can be **provided externally**
- **Further detection** can benefit from this new information

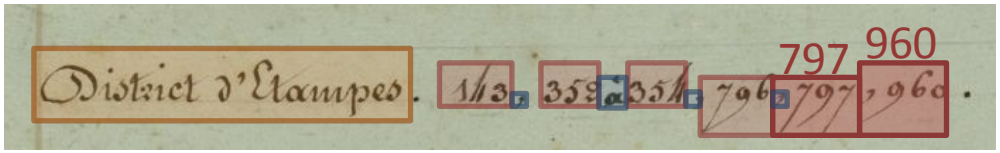
VALIDATION - APPLICATIONS

Experimental setup



Spontaneous interaction
Add a missing separators

VS

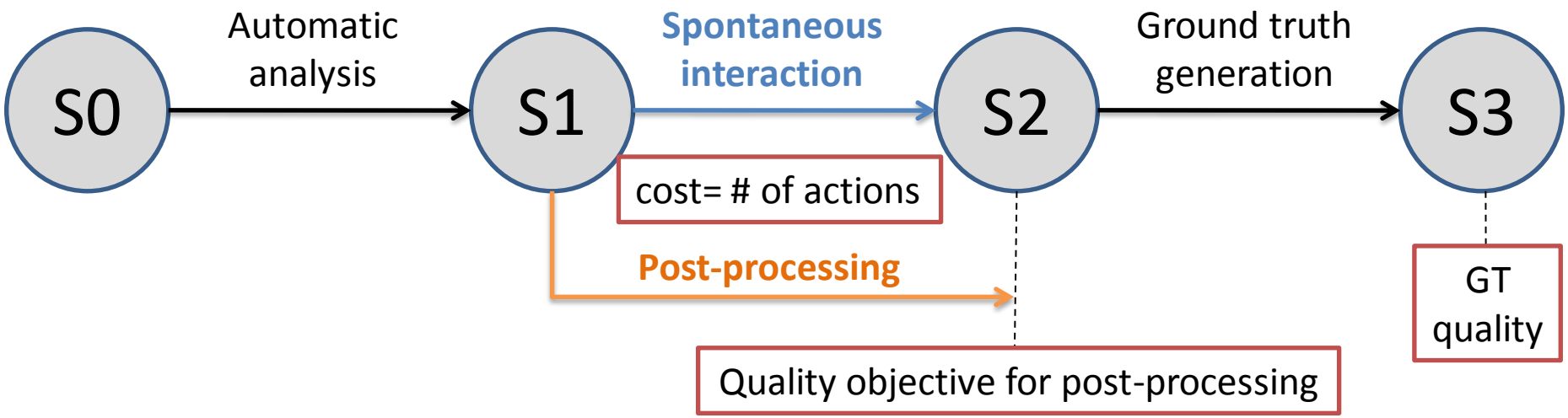


Post processing
Add a zone for each number

Dataset

50 images / 1637 number fields

Protocol



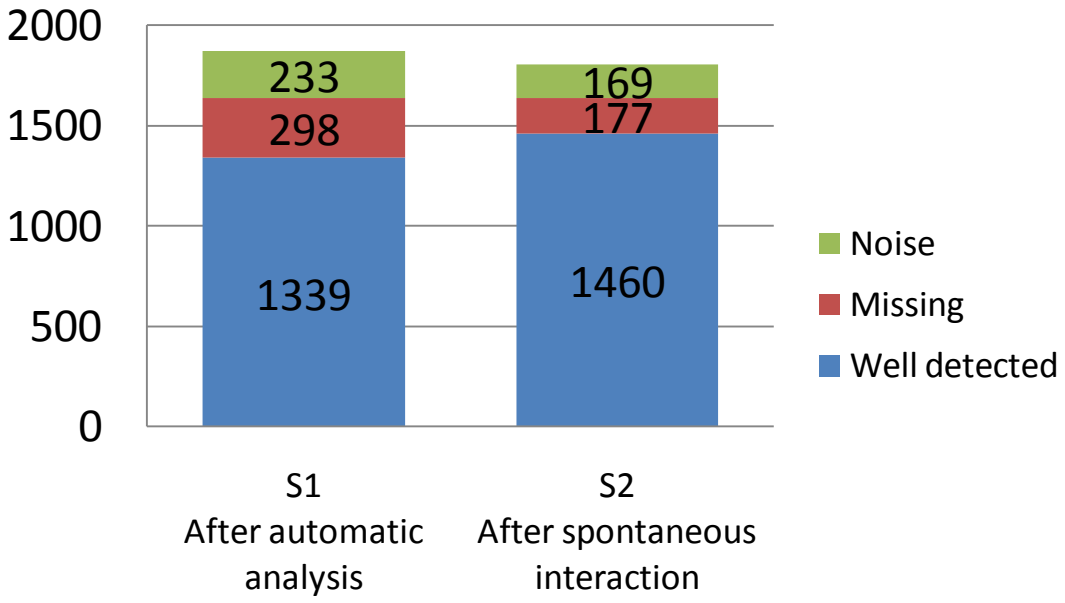
Analysis and results

Analysis

- **Localization quality**: comparison with ground-truth
- **Interaction cost**: # of manual actions required
  - **Our approach** (spontaneous interaction) : # of separators added
  - **Baseline** (post-processing): # of number zones required to reach the same quality level

S1 → S2 quality and cost variations

Quality (# of zones rel. to GT)



Cost to reach S2 quality from S1 (# of actions)

Post-processing (no interaction)	121
With spontaneous interaction	85 (-30%)

CONCLUSION

Spontaneous interaction can be efficient

- Human information should be provided
  - **Asynchronously**
  - **During the analysis**
- When automatic error detection is not possible, **spontaneous interaction is a good backup**
  - **Reduce correction cost** thanks to reprocessing
  - **Easy to implement in an existing system**
    - 4 components
    - Iterative page analyzer
    - Visual Memory
- **Reduction of 30% of number of correction actions** in the experiment shown (vs. post-processing)

Going further

- Quality vs. cost for a complete scenario (ground-truthing)
- Evaluating interaction cost is complex
- Hybrid interactions
- “Interaction” ≠ “Human interaction”